Extracting common sense knowledge from Wikipedia

Suh, S., Halpin, H., & Klein, E. (2006). Extracting common sense knowledge from Wikipedia. In Proceedings of the Workshop on Web Content Mining with Human Language Technologies at ISWC.

Extracting Common Sense Knowledge from Wikipedia

Sangweon Suh, Harry Halpin, and Ewan Klein

School of Informatics
University of Edinburgh
2 Buccleuch Place, Edinburgh, EH8 9LW
S.Suh@sms.ed.ac.uk, H.Halpin@ed.ac.uk, ewan@inf.ed.ac.uk

Abstract. Much of the natural language text found on the web contains various kinds of generic or "common sense" knowledge, and this information has long been recognized by artificial intelligence as an important supplement to more formal approaches to building Semantic Web knowledge bases. Consequently, we are exploring the possibility of automatically identifying "common sense" statements from unrestricted natural language text and mapping them to RDF. Our hypothesis is that common sense knowledge is often expressed in the form of generic statements such as *Coffee is a popular beverage*, and thus our work has focussed on the challenge of automatically identifying generic statements. We have been using the Wikipedia XML corpus as a rich source of common sense knowledge. For evaluation, we have been using the existing annotation of generic entities and relations in the ACE 2005 corpus.

1 Introduction

The Semantic Web's overarching goal, as stated by Tim Berners-Lee, is to enable large-scale creation of machine-readable representations of meaning that make the Web more intelligent and responsive [3]. To date, this has been been driven by formalizing domain-specific ontologies for use with the Semantic Web, such as those produced by the Gene Ontology Consortium.[1] The use of URIs as identifiers makes it possible to connect these diverse ontologies. In practice it is often difficult to merge ontologies in disparate domains. There are several reasons for this, but one of the most obvious is that most ontologies rely upon a large amount of "common sense" background knowledge that has not been explicitly formalized in the ontology. For example, an ontology of food should mention wine, but where is it mentioned in a restaurant ontology that one drinks wine in a restaurant? Or that one can only drink things that are liquids, and that wines are liquids? Common sense knowledge in a Web-accessible format would allow these diverse ontologies to connect, but such a resource does not currently exist. In this paper, we address the question whether it is possible to bootstrap a repository of common sense knowledge on the basis of natural language Web resources such

[1] www.geneontology.org

as Wikipedia. If automatic collation of formalized common sense knowledge is even partially successful, it has the potential to significantly augment the growth of the Semantic Web.

1.1 What is Common Sense?

The concept of "common sense" is ambiguous at best. First, what knowledge qualifies as common sense? While people can make intuitive gradience judgements on whether or not a particular sentence or statement qualifies as common sense, the common sense of a woman from a rural village in Siberia will be very different from a computer programmer in New York City. The original definition of common sense knowledge, as put forward by John McCarthy in his seminal "Programs with Concept Sense" is that "a program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows" [19]. Although problematic, this definition at least gave the initial momentum for the founding artificial intelligence [16]. One of the original proposals was to produce a set of axioms that formalized common sense [14]. The first domain was to be "naive physics", our knowledge of the physical world such as "things that go up must come down". This goal proved unreachable, for the amount of knowledge required was vastly more than the proponents of common sense reasoning expected [23]. Moreover, there was disagreement on the structure and semantics of the knowledge representation language needed to enable common sense reasoning [13]. Recent evidence from cognitive science points out that much of intelligence is based on embodied knowledge that resists easy formalization [23].

More importantly, whether or not *humans* use common sense within a logical framework is irrelevant for the Semantic Web. The question is whether a database of facts that models human common sense knowledge would make computers more easy-to-use and 'intelligent'. There is also good reason to think that the Semantic Web might have a chance of succeeding where artificial intelligence failed [12]. First, the Semantic Web already has a standard formalized semantics for knowledge representation using RDF. Second, the Semantic Web allows, and indeed relies on, decentralized knowledge representation. Third, the Semantic Web has much more modest goals compared with artificial intelligence: instead of making a computer as intelligent as a human being, the Semantic Web initiative only wants to make the Web itself more intelligent than it currently is. A giant, continually-updated database of knowledge that contains a number of common sense facts already exists in the form of Wikipedia. The infrastructure to add RDF to Wikipedia already exists; all that is needed is for triples to be added [26].

However, for engineering purposes, McCarthy's definition of common sense is far too vague. We choose to define "common sense" statements as being explicitly embodied in generic sentences. A necessary, but not sufficient, condition for a clause being classed as generic is that it should contain a generic nominal term as one of the arguments of the main verb. It is usually held that generic nominals refer not to specific entities in the world but rather to a *kind of thing* [5]. This makes them valuable conveyors of conceptual knowledge from an operational

point of view. One important, and easily identified, category of generic nominal is bare plurals (i.e., plural forms without an article): *Foxes gather a wide variety of foods ranging from grasshoppers to fruit and berries.* Additionally, plurals which are preceded by quantifiers such as *every*, *all* and *most* are also likely to be generic.

1.2 Common Sense Research

Given the importance of common sense, it is not surprising that it has been the topic of has been considerable research. The largest project by far is the ***Cyc*** project, which has already collected over a million common sense assertions in two decades [15]. However, these facts are only available for Semantic Web through much smaller ***OpenCyc*** (47,000 concepts and 306,000 assertions) and adding facts to Cyc requires involvement of experts using their own custom knowledge representation language called CycL. Also, upper ontologies are generally not useful. For example, StuffType is not particularly informative. The abstraction of upper ontologies entails a loss of information, and this loss is so severe as to make upper ontologies verge on speculative metaphysics.

A smaller project is the ***OpenMind Commonsense*** [22]. Unlike Cyc, OpenMind capitalizes on the fact that common sense knowledge is "knowledge everyone should know" by collecting it via untrained users over the Web in the form of natural language statements [22]. The project has user-generated 1.6 million natural language statements, and users can add more statements at any time. However, these "facts" are stored in natural language and are of fairly low quality, and attempts like ***ConceptNet*** to use them fail to formally define their relations [17]. A superior method would be to use natural language processing tools to extract statements regularly from a source of updated and monitored natural language statements about the world, as embodied in Wikipedia. This would address the two outstanding problems that need to be dealt with: *extracting a large quantity of statements* and *quality control.*

2 Identifying Generic Statements in Text

Previous work in adapting natural language processing to the Semantic Web has primarily focussed on using these methods to map textual data to preexisting ontologies [18]. Our system is not reliant on regular expressions denoting concepts, like KnowItAll [9], but instead uses hybrid shallow and deep natural language processing.

The method we have adopted for extracting RDF from text can be called 'semantic chunking.' This seems an appropriate term for two reasons. First, we use a syntactic chunker to identify noun groups and verb groups (i.e. nonrecursive clusters of related words with a noun or verb head respectively), after part-of-speech and morphological processing. Second, we use a cascade of finite state rules to map from this shallow syntactic structure into first-order clauses;

this cascade is conceptually very similar to the chunking method pioneered by Abney's Cass chunker [1].

The text processing framework we have used draws heavily on a suite of XML tools developed for generic XML manipulation (LTXML [25]) as well as NLP-specific XML tools (LT-TTT [11], LT-CHUNK [10]). More recently, significantly improved upgrades of these tools have been developed, most notably the program *lxtransduce*, which performs rule-based transductions of XML structures. We have used *lxtransduce* both for the syntactic chunking (based on rules developed by Grover) and for the construction of semantic clauses.

The main steps in the processing pipeline are as follows:

1. Words and sentences are tokenized.
2. The words are tagged for their part of speech using the CandC tagger [8] and the Penn Treebank tagset.
3. The words are then reduced to their morphological stem (lemma) using Morpha [20].
4. The *lxtransduce* program is used to chunk the sentence into verb groups and noun groups
5. In an optional step, words are tagged as Named Entities, using the statistical CandC tagger trained on MUC data.
6. Generic and specific noun groups are identified, using a variety of morphological and lexical cues.
7. The output of the previous step is selectively mapped into semantic clauses in a series of steps, described in more detail below.
8. The XML representation of the clauses is converted using an XSLT stylesheet into RDF and RDFS statements by mapping subject nouns to RDF subjects (with adjective added), verbs to RDF predicates, and objects to RDF objects.

The output of the syntactic processing is an XML file containing word elements which are heavily annotated with attributes. Following CoNLL BIO notation [21], chunk information is recorded at the word level. Heads of noun groups and verb groups are assigned semantic tags such as `arg` and `rel` respectively. In addition, other semantically relevant forms such as conjunction, negation, and prepositions are also tagged. Most other input and syntactic information is discarded at this stage. However, we maintain a record through shared indices of which terms belong to the same chunks. This is used, for instance, to build coordinated arguments.

Regular expressions over the semantically tagged elements are used to compose clauses, using the heuristic that an `arg` immediately preceding a `pred` is the subject of the clause, while `arg`s following the `pred` are complements. Since the heads of verb groups are annotated for voice, we can treat passive clauses appropriately, yielding a representation that is equivalent to the active congener. We also implement simple heuristics that allow use to capture simple cases of control and verb phrase ellipsis in many cases.

2.1 Evaluation: Wikipedia corpus

The automatic creation of semantic metadata could bring relevant facts and relationships to the attention of a human ontology creator that might otherwise be lost among large corpora of texts [7]. This could be especially useful when trying to annotate the Semantic Wikipedia for common sense facts [26].

There is no obvious methodology for evaluating the extraction of RDF triples from Wikipedia. The authors assessed the semantic utility (whether or not a statement counted as "sensical" and was "meaning-preserving" from the original Wikipedia page) of 585 triples in randomly processed 200 Wikipedia pages and two judges who carried out the assessment achieved a κ score of .41, showing their agreement on what constituted "common-sense" knowledge to be moderate. The average percentage correct was 51%. The marking results of the judges are given in Table 1, where the column is the results of one judge and the row those of the second judge.

Table 1. Distribution of Ratings of Wikipedia Triples

Marking	**Yes**	**No**
Yes	209	126
No	48	202

From inspection of Table 1 it should become clear that the results are further explained by the fact that one judge was much more careful about his ascription of "common-sense" knowledge to triples than the other judge, as he marked triples "No" that the other judge marked "Yes" three times more often than the reverse situation. The extracted triples were also judged on whether or not they produced actual generic knowledge by one judge, who found that at least 14% (82 out of 587) of the triples were definitely not generics.

While these results seem unimpressive, we still believe even automatically creating for every web page an average of $2 \sim 3$ triples which are approximately half correct, would still be a boon to people attempting to annotate Wikipedia with knowledge in RDF.

2.2 Evaluation: ACE Corpus

An evaluation was performed to estimate how well the generic noun-group labels produced match Gold Standard labels. The ACE Multilingual Training Corpus 2005 includes annotation of generic entities and events, and we used 412 files in their broadcast news, news wire and web log sections, which account for 55% of the whole corpus. Table 2 gives the results.

The ACE annotation scheme [2] defines five classes of entities: negatively quantified (NEG), attributive (ATR), specific referential (SPC), generic referential (GEN), under-specified referential USP). An entity is SPCwhen the entity being referred to is a particular, unique object (or set of objects): (e.g.) *[John's lawyer]*

won the case. ATR indicates that the entity is only being used to attribute some property or attribute to some entity, such as in *John is [a lawyer].* Finally, the USP class is reserved for quantified and ambiguous NPs which the annotators could not confidently assign to the other four classes.

Table 2. Labeling precision against ACE entity class annotations

entity class	SPC	GEN	USP	NEG	ATR
ng-gen	34.2%	28.9%	36.8%	0.1%	0.0%

(total 1244 ng-gen terms)

In the results shown in Table 2, if we only accept exact matches with GEN class, the labeling precision is 28.9%. But since our our operational definition of generics, based on morpho-syntactic features also encompasses many terms that are labeled as USP in ACE, it would be more appropriate to consider matches with GEN and USP merged together. This gives the higher precision figure of 65.7%. As further justification of this merging of classes, note that it accords with our primary objective of whether we can attain reasonable discrimination against specific-referential (SPC) entities.

Among the 1244 generic noun groups extracted by our method, about one third were in fact classified as SPC by the ACE annotators. That is, even though those terms are not definite constructions, they were judged to be referring to specific entities in their discourse context. Error analysis reveals clearly the difficulties. As an illustration, consider the following mismatch case.

> Some 70 people were arrested Saturday as *demonstrators* clashed with *police* at the end of a major peace rally here. ...

In the above context, both *demonstrators* and *police* are SPC entities according to ACE, as they refer to a particular set of people who actually participated in the stated event. However, in another context, such as the one below, the same terms would be correctly classed as generic by our method:

> *Demonstrators* are often aggressive to *police*.

Among the mismatch results of Table 2, most of the true SPC cases can be viewed similarly. The difficulties of this kind suggest us to do some sentence level processing as next step for systematic improvement. We could make use of time-space anchor words within a sentence, such as *Saturday* and *here* in the above example. In addition, taken together with these anchor words, the head verb *clash* in past tense would indicate that the sentence describes an episodic event rather than a generic event. These points need to be implemented as preconditions that prevent the two terms *demonstrators* and *police* from being classed as GEN.

As a further analysis based on ACE, we also made an explicit comparison between bare plurals and bare singulars(i.e. singular nouns without an article)

with regard to their association with generic terms. It turned out that bare plurals occurred in text about two times more frequently than bare singulars and that the labeling precision was markedly lower for bare singulars compared to bare plurals (33.3% vs. 79.4%); bare singulars had a strong tendency to be SPCs.

3 Semantic Representation of Generics

3.1 Logical Representation in RDF(S)

There are problems with selecting suitable subject/object arguments because the arguments in the relations are often composite in structure and associated with prepositional phrases. The argument preceding a predicate is turned into the subject of an RDF triple and the argument immediately following the predicate into the object. As we are focussing on generic sentences, we filter out clauses where none of the arguments are identified as generic.

In the following, we illustrate our approach with a set of RDFS statements (in N-Triples format) extracted from a sample Wikipedia article. The last four triples in this set result from the corresponding generic sentences in the article, as identified by the method discussed above:

- Rodents lack canines, and have a space between their incisors and premolars.
- Rodents have two incisors in the upper as well as in the lower jaw ...
- The earliest rodents resembled squirrels and from these stem rodents, they diversified.
- During the Pliocene, rodent fossils appeared in Australia.

For simplicity, we map a head noun of the argument into an RDFS class (an instance of `rdfs:Class`) and the predicate of relation into an RDF property (an instance of `rdf:Property`). For class names, we use the word's lemma as the argument content, which may be attached to any contingent modifier words returned by our semantic interpretation module.[2]

```
wiki:rodent  rdf:type  rdfs:Class .
wiki:canine  rdf:type  rdfs:Class .
wiki:incisor  rdf:type  rdfs:Class .
wiki:early_rodent  rdf:type  rdfs:Class .
wiki:squirrel  rdf:type  rdfs:Class .
wiki:rodent_fossils  rdf:type  rdfs:Class .

wiki:lack  rdf:type  rdf:Property .
wiki:have  rdf:type  rdf:Property .
wiki:resemble  rdf:type  rdf:Property .
```

[2] Strictly speaking, we should replace a triple such as `wiki:rodent wiki:lack wiki:canine` by the RDFS statements `wiki:lack rdfs:domain wiki:rodent` and `wiki:lack rdfs:range wiki:canine`, but we use the RDF syntax for convenience.

```
wiki:appear_in  rdf:type  rdf:Property .

wiki:rodent  wiki:lack  wiki:canine .
wiki:rodent  wiki:have  wiki:incisor .
wiki:early_rodent  wiki:resemble  wiki:squirrel .
wiki:rodent_fossils  wiki:appear_in  wiki:Australia .
```

Terms like `early_rodent` and `rodent_fossils` are constructed at different stages in the processing pipeline: complex nominals (*rodent fossils*) are combined into semantic primitives (`rodent_fossils`) once the noun group chunking results are available, while the attachment of adjectival modifiers (*early rodents*) is carried out in the final stylesheet processing for RDF transformation. As regards the `rodent_fossils`, it was compounded as two consecutive nominals were matched within the same noun group; this can be verified through an attribute added by the pipeline on construction of compound nominals.

The statements are extracted from sentences that have the generic noun groups either as subject or object argument, and we can see that they contain plausible knowledge about the domain of rodents. Our goal is to scale-up the extraction to process a much larger set of sentences, leading to connections between the triples which will achieve the effect of knowledge integration.

3.2 Issues in RDF(S) Representation of Generics

Not surprisingly, it is difficult to represent the semantics of generics in as simple a framework as RDFS. The semantic interpretation of generics has received copious discussion in the linguistics and philosophy literature [5, 6]. The primary challenge is that generics have the characteristics of default statements, in the sense that generics admit exceptions, and this is entirely lacking from our current representation. For example, the generic statement *dogs bark* is not rendered false by the fact that some dogs fail to bark; in RDFS, either we should rule out existence of any dog that does not bark, or we cannot group all individual dogs into an RDFS class representing the dog genus that has a property of barking. Conversely, when exemplars of a kind are scarce, the fact that a few exemplars share a property P does not warrant a generic statement that the kind as a whole has property P. Although we do not have a proposal for dealing with this aspect of generics, given the relative lack of agreement about mechanisms for performing logical inference in the Semantic Web, this does not seem too worrying as we can restrict ourselves to mining data sources like Wikipedia that contain relatively uncontroversial statements (most of the time).

As is well known, there are other significant limitations to RDF(S). The semantics of RDF does not assign any logical role to negation, so we cannot adequately express statements like *Foxes are not pack animals*. Moreover, we cannot define a class like `rodent_fossils` as the intersection of the classes `rodent` and `fossils`. These shortcomings provide motivation for using OWL DL as a representation framework in place of RDF(S).

4 Conclusion and Future Work

This research is still in its early stages, but we believe it offers an innovative way of contributing to the construction of ontologies for the Semantic Web by virtue of its linguistic focus on generics, since these map naturally towards the types of information embodied in ontologies. Likewise, we believe the prospect of using this methodology to help generate semantic annotation for the Semantic Wikipedia project is both exciting and useful. One future avenue to explore is whether next step is to investigate how the combination of human validation and automatic semantic annotation can improve the work through active-learning [24].

There is a continuum of approaches for carrying out the kind of task we are exploring here. At one pole are very shallow techniques for extracting rather impoverished semantic information, while at the other extreme, one can use deep statistical parsers [4] to build much richer semantic structures. Since all wide-coverage techniques are error-prone, we believe that the most urgent task is to find the right balance between accuracy and volume in knowledge extraction techniques, and develop semantically-oriented techniques for cleaning noisy semantic data. In addition, as already indicated above, we believe it would be helpful to go beyond RDF(S) by mapping directly to an OWL DL ontology.

By using human language technologies to semantically annotate Wikipedia, we show that utilizing the collective intelligence of ordinary users of the Web can provide a way to bootstrap tremendous amounts of common sense data that otherwise would take decades to engineer. In concert with human validation to prune out mistakes, the dream of a fully-annotated Semantic Wikipedia could become more feasible. An annotated Semantic Wikipedia would lead in turn to a greater *network effect* for the Semantic Web, so the Semantic Web would be used not only as a tool to reason about specialized ontologies, but also to reason about the myriad facts and vagaries that make up everyday life.

References

1. S. Abney. Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344, 1996.
2. Ace (automatic content extraction) english annotation guidelines for entities version 5.6.1 (2005.05.23), 2005.
3. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.
4. J. Bos, S. Clark, M. Steedman, J. Curran, and J. Hockenmaier. Wide-coverage semantic representations from a CCG parser. In *In Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, pages 1240–1246, Geneva, Switzerland, 2004.
5. G. N. Carlson. Generic terms and generic sentences. *Journal of Philosophical Logic*, 11(2):145–181, 1982.
6. G. N. Carlson and F. J. Pelletier, editors. *The Generic Book*. University of Chicago Press, 1995.

7. P. Cimiano and J. Volker. Text2Onto — a framework for ontology learning and data-driven change discovery. In *Proceedings of International Conference on Applications of Natural Language to Information Systems (NLDB'05)*, 2005.
8. S. Clark and J. Curran. Parsing the WSJ using CCG and log-linear models. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, pages 104–111, Barcelona, Spain, 2004.
9. O. Etzioni, M. J. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. Web-scale information extraction in know-it-all. In *Proceedings of World Wide Web Conference (WWW 2004)*, 2004.
10. S. Finch and A. Mikheev. A workbench for finding structure in texts. In W. Daelemans and M. Osborne, editors, *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP-97)*. Washington D.C., 1997.
11. C. Grover, C. Matheson, A. Mikheev, and M. Moens. Lt ttt—a flexible tokenisation tool. In *LREC 2000—Proceedings of the 2nd International Conference on Language Resources and Evaluation*, pages 1147–1154, 2000.
12. H. Halpin. The Semantic Web: The Origins of Artificial Intelligence Redus. In *Third International Workshop on the History and Philosophy of Logic, Mathematics, and Computation*, San Sebastian, Spain, 2004.
13. P. Hayes. In defense of logic. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 559–565. William Kaufmann, Cambridg, MA, 1977.
14. P. Hayes. The second naive physics manifesto. In *Formal Theories of the Commonsense World*. Ablex, 1986.
15. D. Lenat. Cyc: Towards Programs with Common Sense. *Communcations of the ACM*, 33(8):30–49, 1990.
16. D. Lenat and E. Feigenbaum. On the Thresholds of Knowledge. In *In Proceedings of International Joint Conference on Artificial Intelligence*. William Kaufmann, Cambridg, MA, 1987.
17. H. Liu and P. Singh. ConceptNet: A lexical database for english. *BT Technology Journal*, 4(22):211–226, 2004.
18. LREC. *Towards a Language Infrastructure for the Semantic Web*, Lisbon, Portugal, 2004.
19. J. McCarthy. Programs with common sense, 1959. http://www-formal.stanford.edu/jmc/mcc59.html.
20. J. C. Minnen, G. and D. Pearce. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–203, 2001.
21. E. F. T. K. Sang and S. Buchholz. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of the Conference on Natural Language Learning (CoNLL-2000)*. Lisbon, Portugal, 2000.
22. P. Singh. The public acquisition of commonsense knowledge. In *In Proceedings of AAAI Spring Symposium on Acquiring (and Using) Linguistic (and World) Knowledge for Information Access*. AAAI, Palo Alto, CA, 2002.
23. B. C. Smith. The Owl and the Electric Encyclopedia. *Artificial Intelligence*, 47:251–288, 1991.
24. C. Thompson, M. Califf, and R. Mooney. Active learning for natural language parsing and information extraction. In *In Proceedings of the 16th InternationalMachine Learning Conference (ICML 1999)*, pages 406–414, Bled, Slovenia, 1999.
25. H. Thompson, R. Tobin, D. McKelvie, and C. Brew. LT XML. software API and toolkit for XML processing, 1997.
26. M. Völkel, D. V. M. Krötzsch, H. Haller, and R. Studer. Semantic wikipedia. In *In Proceedings of the 15th International World Wide Web Conference (WWW 2006)*, pages 585–594, Edinburgh, Scotland, 2006.